

Práctica Nro. 5

Asignatura: Tecnologías de Información

Ejercicio:

1. Considere el siguiente esquema relacional:

Empleado(idEmpleado, nombreEmpleado, fechaIngreso)
Servicio (idServicio, nombreServicio, valorServicio)
vehiculo(idVehiculo, marcaVehiculo, modeloVehiculo)
prestación_servicio(id_ps, idServicio, idEmpleado, idVehiculo, fecha)
FK: idServicio->servicio, idEmpleado->Empleado, idVehiculo->vehiculo

- a) Usando SQL crear la tabla para el sistema de Prestación de Servicios del taller automotriz

```
CREATE TABLE Servicio (  
idServicio int(4) UNSIGNED ZEROFILL NOT NULL,  
nombreServicio varchar(50),  
valorServicio int,  
PRIMARY KEY (idServicio)  
);
```

```
CREATE TABLE Empleado (  
idEmpleado int(3) UNSIGNED ZEROFILL NOT NULL ,  
nombreEmpleado varchar(50),  
fechaIngreso date,  
PRIMARY KEY (idEmpleado)  
);
```

```
CREATE TABLE Vehiculo (  
idVehiculo char(6) NOT NULL,  
marcaVehiculo varchar(20),  
modeloVehiculo varchar(20),  
PRIMARY KEY (idVehiculo)  
);
```

```
CREATE TABLE prestacionServicio (  
idPs int(8) AUTOINCREMENT,  
idServicio int(4) UNSIGNED ZEROFILL NOT NULL,  
idEmpleado int(3) UNSIGNED ZEROFILL NOT NULL ,  
idVehiculo char(6) NOT NULL,  
fecha date,
```

PRIMARY KEY (idPs),
FOREIGN KEY (idServicio) REFERENCES Servicio,
FOREIGN KEY (idEmpleado) REFERENCES Empleado,
FOREIGN KEY (idVehiculo) REFERENCES Vehiculo);

- b) Inserte los datos del empleado Mario Molina que ingresó a la organización el 10/02/2010

```
INSERT INTO Empleado( idEmpleado, nombreEmpleado, fechaIngreso)
VALUES (1, "Mario Molina", 10/02/2010);
```

o bien,

```
INSERT INTO Empleado
VALUES (1, "Mario Molina", 10/02/2010);
```

- c) Mostrar la cantidad de prestaciones de servicio ejecutadas entre el 01 de octubre del 2018 y el 23 de noviembre del 2018.

```
SELECT COUNT(*)
FROM prestacionServicio
WHERE fecha BETWEEN 01/10/2018 AND 23/11/2018;
```

- d) Mostrar la cantidad total de prestaciones realizadas agrupadas por idVehiculo

```
SELECT idVehiculo, COUNT(idServicio)
FROM prestacionServicio
GROUP BY idVehiculo;
```

- e) Mostrar los vehículos con la menor cantidad de prestaciones de servicio realizadas.

```
SELECT idVehiculo, COUNT(idServicio)
FROM prestacionServicio
GROUP BY idVehiculo
ORDER BY COUNT(idServicio) ASC;
```

- f) Mostrar el nombre del empleado, nombre del Servicio y modelo del vehículo en la prestación de servicio N°7 (idPs)

```
SELECT Empleado.nombreEmpleado, Servicio.nombreServicio,
Vehiculo.modeloVehiculo
FROM Empleado, Servicio, Vehiculo, prestacionServicio
WHERE Empleado.idEmpleado = prestacionServicio.idEmpleado
AND Servicio.idServicio = prestacionServicio.idServicio
AND Vehiculo.idVehiculo = prestacionServicio.idVehiculo;
AND prestacionServicio.idPs = 7;
```

- g) Como quedaría el SQL de la tabla prestacionServicio, si es que: la restricción de las claves foráneas no se realiza con el método RESTRICT(recuerde que por defecto se realiza este tipo de restricción) , sino que cuando se haga un update se realizan con método CASCADE y cuando se haga un DELETE se dejan en NULL, salvo para el idServicio, que se quiere que en caso de que se borre un servicio, las referencias a este queden en 9999.

```
CREATE TABLE prestacionServicio (  
  id_ps int,  
  idServicio int DEFAULT 9999,  
  idEmpleado int,  
  idVehiculo int,  
  fecha date,  
  PRIMARY KEY (id_ps)  
  FOREIGN KEY (idServicio) REFERENCES Servicio  
    ON UPDATE CASCADE  
    ON DELETE SET DEFAULT  
  FOREIGN KEY (idEmpleado) REFERENCES Empleado  
    ON UPDATE CASCADE  
    ON DELETE SET NULL  
  FOREIGN KEY (idVehiculo) REFERENCES vehiculo)  
    ON UPDATE CASCADE  
    ON DELETE SET NULL
```

- h) Como ejercicio de estudio ingrese datos a todas las tablas, para que pueda comprobar empíricamente todas las consultas y restricciones puestas a las claves foráneas.